



6.867 Machine learning: lecture 1

Tommi S. Jaakkola

MIT CSAIL

tommi@csail.mit.edu



6.867 Machine learning: administrivia

- Course staff (6867-staff@lists.csail.mit.edu)
 - Prof. Tommi Jaakkola (tommi@csail.mit.edu)
 - Adrian Corduneanu (adrianc@mit.edu)
 - Biswajit (Biz) Bose (cielbleu@mit.edu)
- General info
 - lectures MW 2.30-4pm in 32-141
 - tutorials/recitations, initially F11-12.30 (4-145) / F2.30-4 (4-159)
 - website <http://www.ai.mit.edu/courses/6.867/>
- Grading
 - midterm (15%), final (25%)
 - 5 (\approx bi-weekly) problem sets (30%)
 - final project (30%)

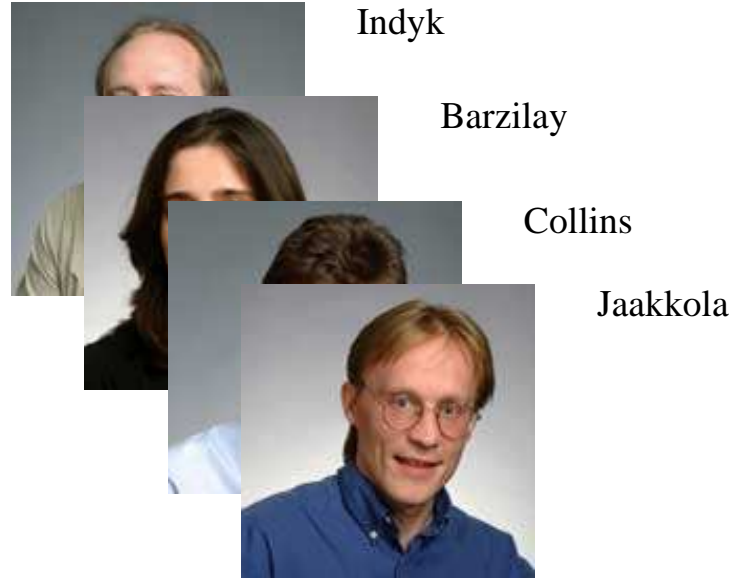


Why learning?

- Example problem: face recognition

Why learning?

- Example problem: face recognition



Training data: a collection of images and labels (names)

Why learning?

- Example problem: face recognition



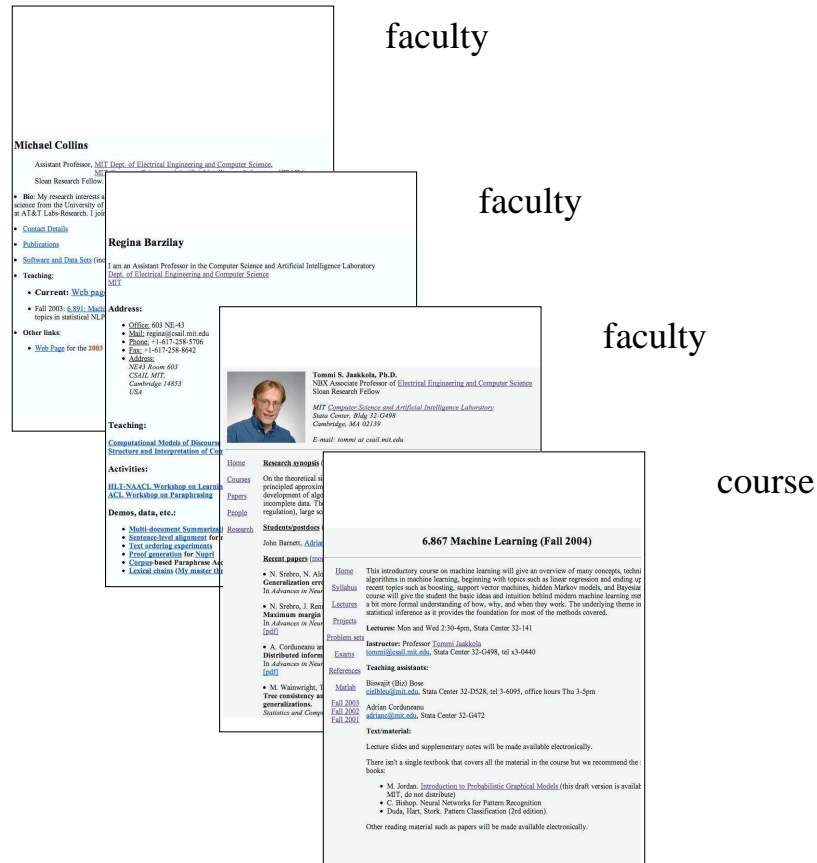
Training data: a collection of images and labels (names)



Evaluation criterion: correct labeling of new images

Why learning?

- Example problem: text/document classification



The collage consists of several overlapping webpages:

- Faculty pages:**
 - Michael Collins:** Assistant Professor, MIT Dept. of Electrical Engineering and Computer Science. Sloan Research Fellow. Bio: My research interests are in machine learning from the University of Cambridge and AT&T Labs-Research. Contact Details, Publications, Software and Data Sets (links), Teaching.
 - Regina Barzilay:** Assistant Professor in the Computer Science and Artificial Intelligence Laboratory, MIT. Bio: I am an Assistant Professor in the Computer Science and Artificial Intelligence Laboratory, MIT. Current: Web page. Fall 2003: 6.867: Machine Learning in Statistical NLP. Other links: Web Page for the 2003.
 - Tommi S. Jaakkola, Ph.D.:** NXY Associate Professor of Electrical Engineering and Computer Science, Sloan Research Fellow. MIT Computer Science and Artificial Intelligence Laboratory, State Center Bldg. 32-G498, Cambridge, MA 02139. E-mail: tommi@csail.mit.edu. Teaching: Computational Models of Discourse Structure and Interpretation of Complex Texts. Research examples: HLT-NAACL Workshop on Learning NLP, Workshop on Empowering Demos, data, etc. Multi-document Summarization, Sentence-level alignment for Text mining experiments, Text generation for Signet, Corpus-based Paraphrase Mining, Lexical chains. Other links: A. Coruhoglu and Distributed inference in Adaboost on NLP, M. Wainwright, Tree consistency as generalization, Statistics and Comp.
- Course page:**
 - 6.867 Machine Learning (Fall 2004)**
 - Home:** This introductory course on machine learning will give an overview of many concepts, technical algorithms in machine learning, beginning with topics such as linear regression and ending at recent topics such as boosting, support vector machines, hidden Markov models, and Bayesian networks. This course will give the student the basic ideas and intuition behind modern machine learning methods but not a more formal understanding of how, why, and when they work. The underlying theme is statistical inference as it provides the foundation for most of the methods covered.
 - Lectures:** Mon and Wed 2:30-4pm, State Center 32-141
 - Instructor:** Professor Tommi Jaakkola, tommi@csail.mit.edu, State Center 32-G498, tel 43-0440
 - Teaching assistants:**
 - Blaschke (Bliz) Bose, bliz@csail.mit.edu, State Center 32-D528, tel 3-6095, office hours Thu 3-5pm
 - Adrian Coruhoglu, acoruh@csail.mit.edu, State Center 32-G472
 - Text/material:** Lecture slides and supplementary notes will be made available electronically. There isn't a single textbook that covers all the material in the course but we recommend the following books:
 - M. Jordan, *Introduction to Probabilistic Graphical Models* (this draft version is available at MIT, do not distribute)
 - C. Bishop, *Neural Networks for Pattern Recognition*
 - Duda, Hart, Stork, *Pattern Classification (2nd edition)*.Other reading material such as papers will be made available electronically.

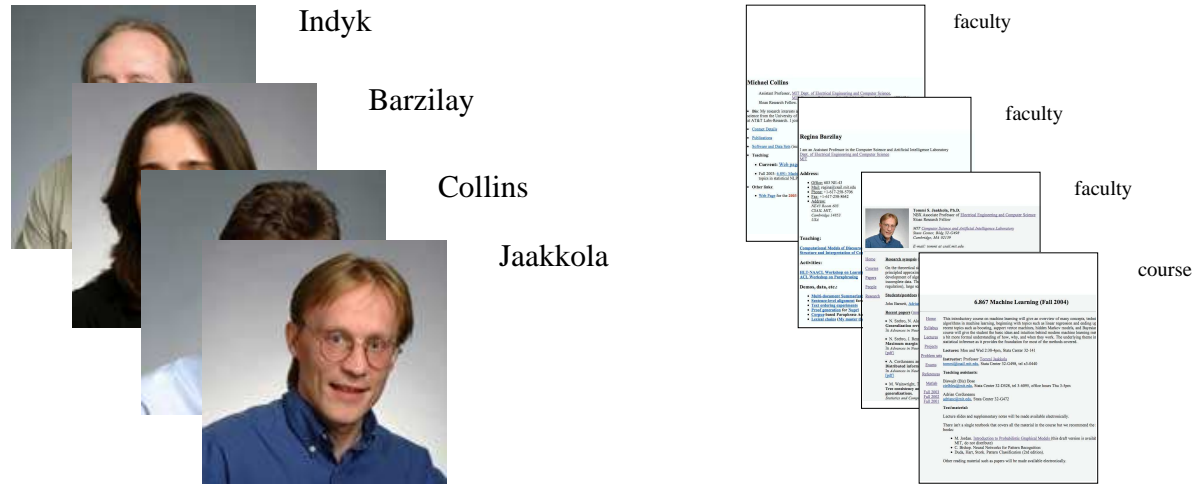
- a few labeled training documents (webpages)
- goal to label yet unseen documents



Why learning?

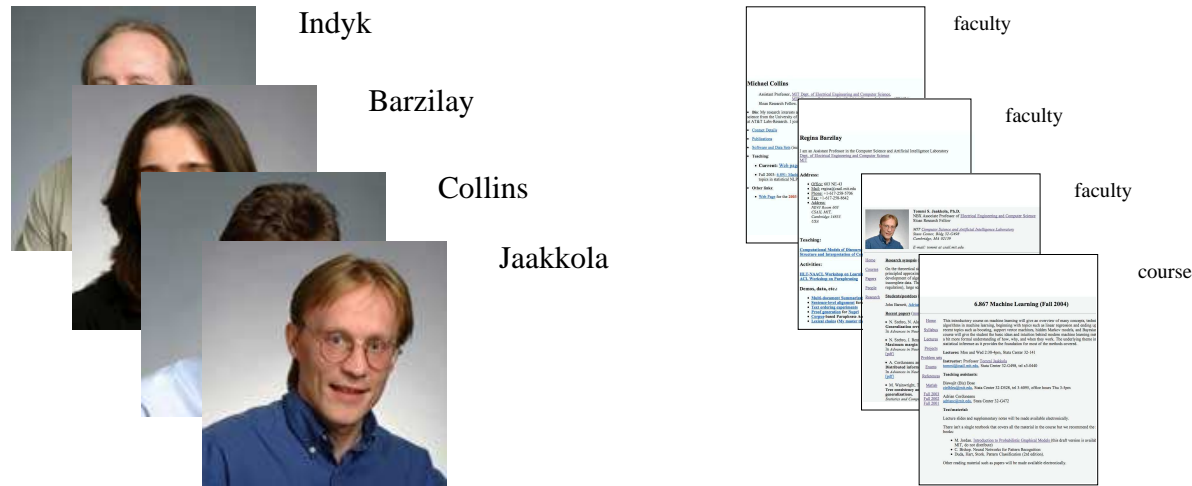
- There are already a number of applications of this type
 - face, speech, handwritten character recognition
 - fraud detection (e.g., credit card)
 - recommender problems (e.g., which movies/products/etc you'd like)
 - annotation of biological sequences, molecules, or assays
 - market prediction (e.g., stock/house prices)
 - finding errors in computer programs, computer security
 - defense applications
 - etc

Learning



- Steps
 - entertain a (biased) set of possibilities (hypothesis class)
 - adjust predictions based on available examples (estimation)
 - rethink the set of possibilities (model selection)

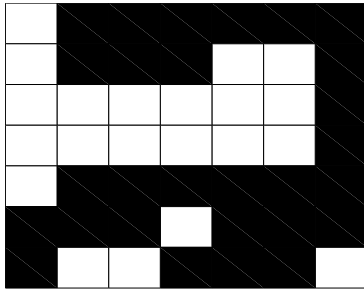
Learning



- Steps
 - entertain a (biased) set of possibilities (hypothesis class)
 - adjust predictions based on available examples (estimation)
 - rethink the set of possibilities (model selection)

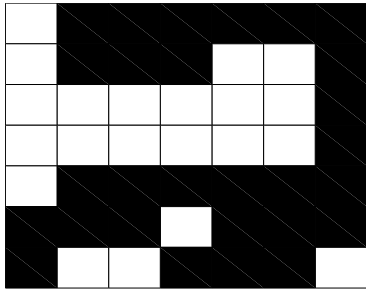
- Principles of learning are “universal”
 - society (e.g., scientific community)
 - animal (e.g., human)
 - machine

Learning, biases, representation

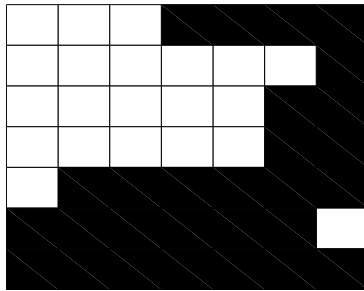


“yes”

Learning, biases, representation

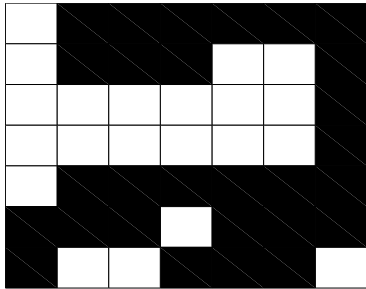


“yes”

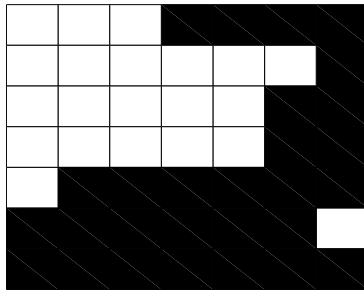


“yes”

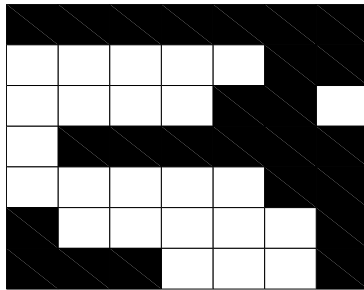
Learning, biases, representation



“yes”

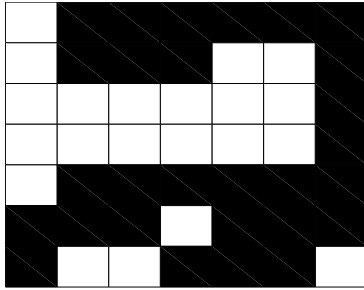


“yes”



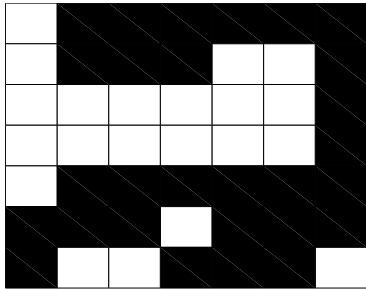
“no”

Learning, biases, representation

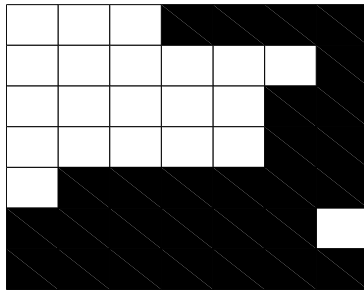


“yes”

Learning, biases, representation

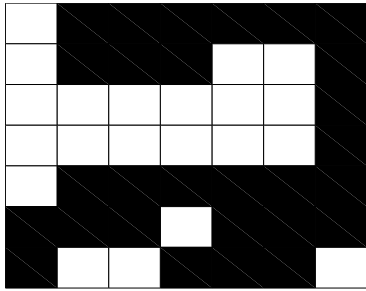


“yes”

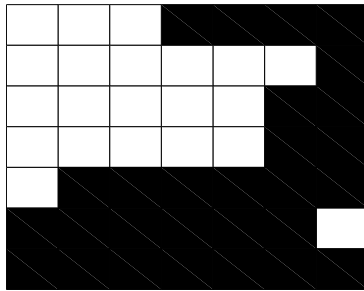


“yes”

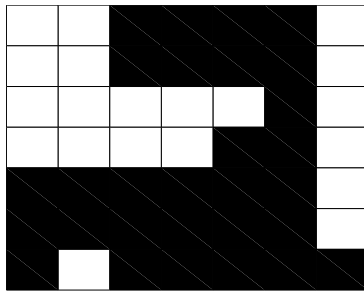
Learning, biases, representation



“yes”



“yes”

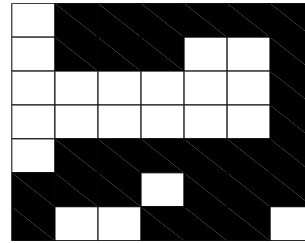


“no”

(oops)

Representation

- There are many ways of presenting the same information



0111111001110010000000100000001001111110111011111001110111110001

- The choice of representation may determine whether the learning task is very easy or very difficult



Representation

0111111001110010000000100000001001111110111011111001110111110001
0001111100000011000001110000011001111110111111001111111101111111
1111111000000110000011000111111000000111100000111110001101110001

“yes”

“yes”

“no”

Representation

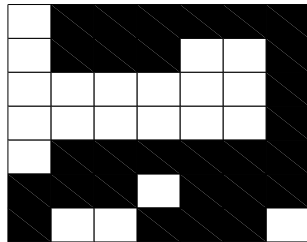
```

0111111001110010000000100000001001111110111011111001110111110001
0001111100000011000001110000011001111110111111001111111101111111
1111111000000110000011000111111000000111100000111110001101110001
  
```

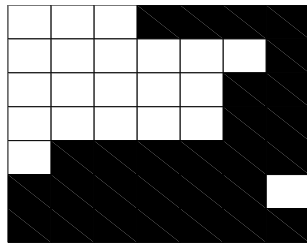
“yes”

“yes”

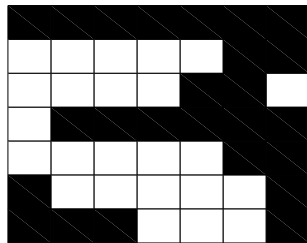
“no”



“yes”



“yes”

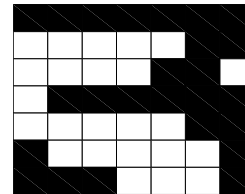


“no”

Hypothesis class

- Representation: examples are binary vectors of length $d = 64$

$$\mathbf{x} = [111 \dots 0001]^T =$$



and labels $y \in \{-1, 1\}$ (“no”, “yes”)

- The mapping from examples to labels is a “linear classifier”

$$\hat{y} = \text{sign}(\theta \cdot \mathbf{x}) = \text{sign}(\theta_1 x_1 + \dots + \theta_d x_d)$$

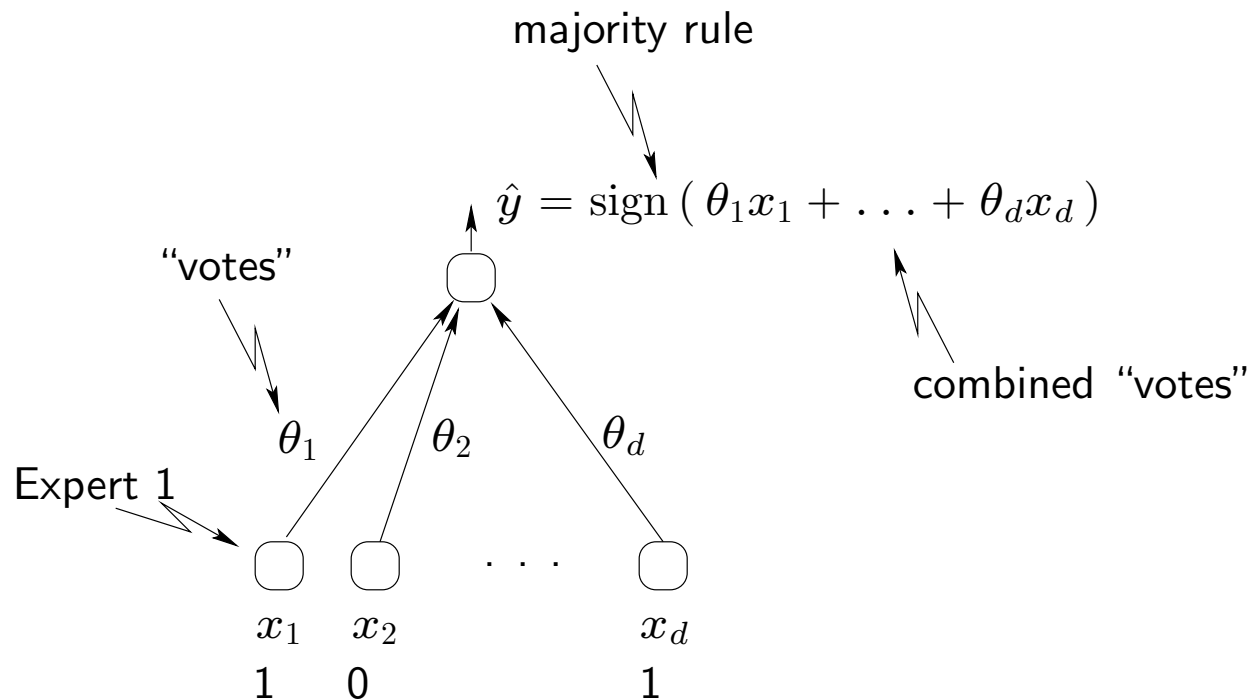
where θ is a vector of *parameters* we have to learn from examples.

Linear classifier/experts

- We can understand the simple linear classifier

$$\hat{y} = \text{sign}(\theta \cdot \mathbf{x}) = \text{sign}(\theta_1 x_1 + \dots + \theta_d x_d)$$

as a way of combining expert opinion (in this case simple binary features)



Estimation

\mathbf{x}	y
011111100111001000000010000000100111111011101111110011101111110001	+1
0001111100000011000001110000011001111110111111001111111100000011	+1
1111111000000110000011000111111000000111100000111110001101111111	-1
...	...

- How do we adjust the parameters θ based on the labeled examples?

$$\hat{y} = \text{sign}(\theta \cdot \mathbf{x})$$

Estimation

\mathbf{x}	y
0111111001110010000000100000001001111110111011111001110111110001	+1
0001111100000011000001110000011001111110111111001111111100000011	+1
1111111000000110000011000111111000000111100000111110001101111111	-1
...	...

- How do we adjust the parameters θ based on the labeled examples?

$$\hat{y} = \text{sign}(\theta \cdot \mathbf{x})$$

For example, we can simply refine/update the parameters whenever we make a mistake:

$$\theta_i \leftarrow \theta_i + y x_i, \quad i = 1, \dots, d \quad \text{if prediction was wrong}$$

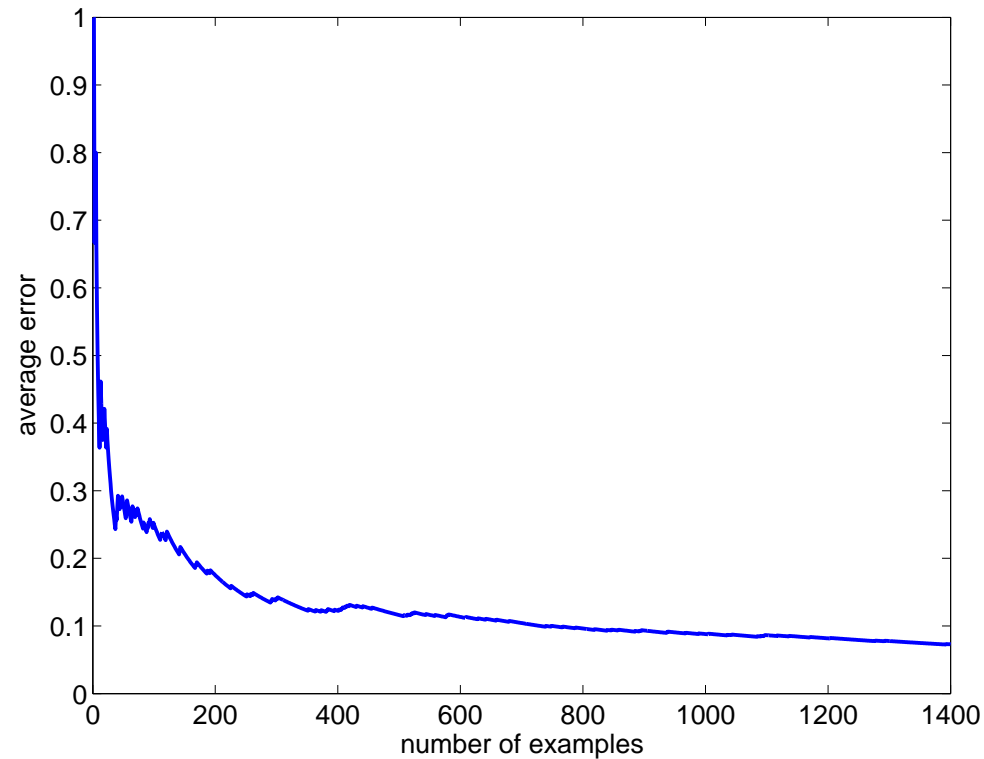


Evaluation

- Does the simple mistake driven algorithm work?

Evaluation

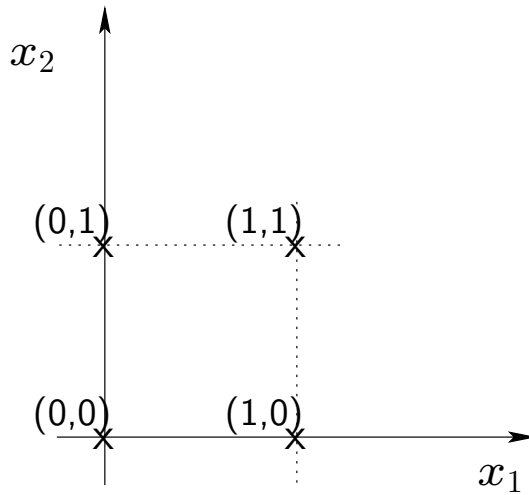
- Does the simple mistake driven algorithm work?



(average classification error as a function of the number of examples and labels seen so far)

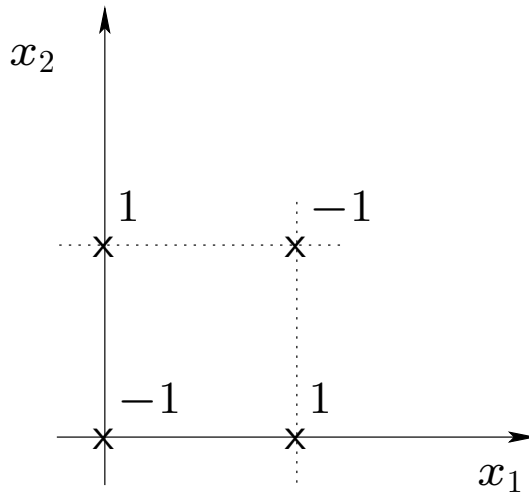
Model selection

- The simple linear classifier cannot solve all the problems (e.g., XOR)



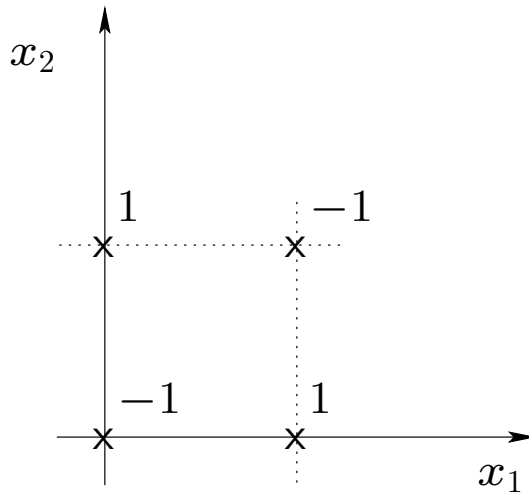
Model selection

- The simple linear classifier cannot solve all the problems (e.g., XOR)



Model selection

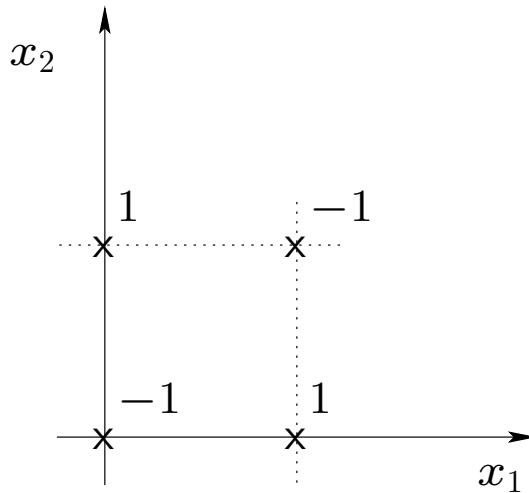
- The simple linear classifier cannot solve all the problems (e.g., XOR)



- Can we rethink the approach to do even better?

Model selection

- The simple linear classifier cannot solve all the problems (e.g., XOR)

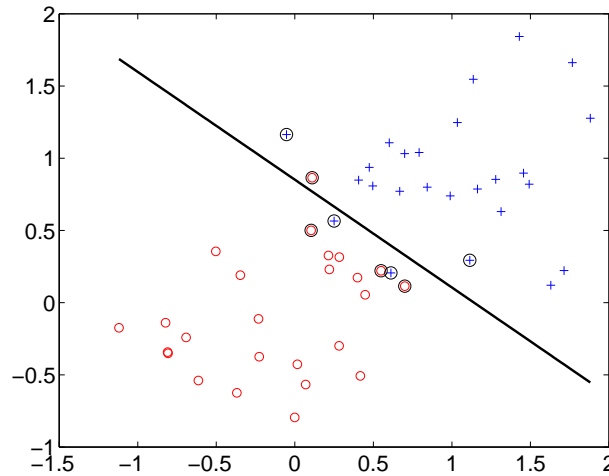


- Can we rethink the approach to do even better?

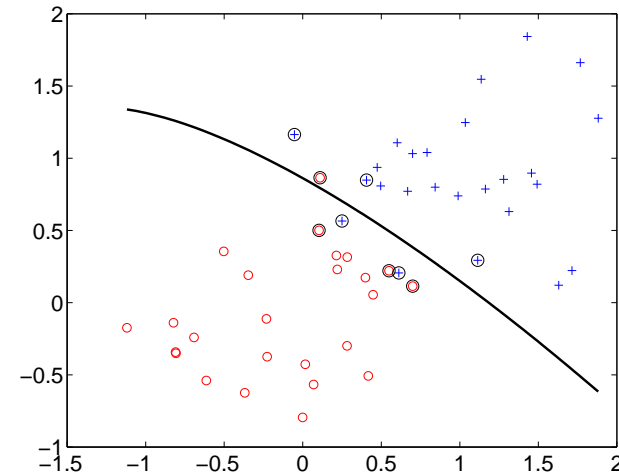
We can, for example, add “polynomial experts”

$$\hat{y} = \text{sign}(\theta_1 x_1 + \dots + \theta_d x_d + \theta_{12} x_1 x_2 + \dots)$$

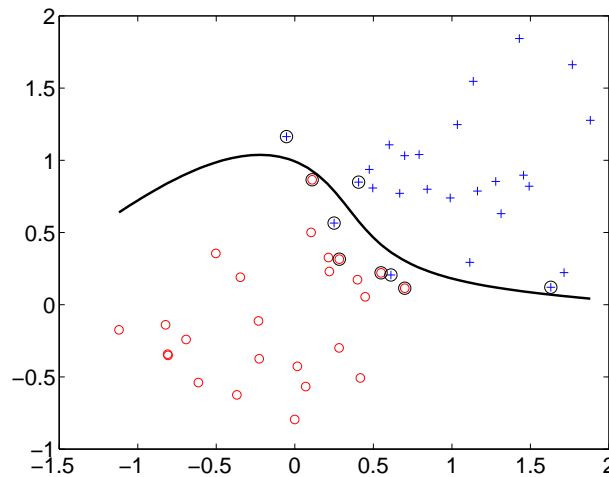
Model selection cont'd



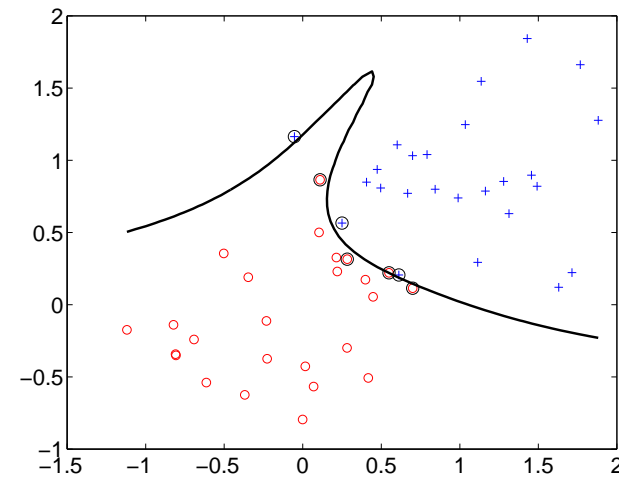
linear



2nd order polynomial



4th order polynomial



8th order polynomial



Types of learning problems (not exhaustive)

- *Supervised* learning: explicit feedback in the form of examples and target labels
 - goal to make predictions based on examples (classify them, predict prices, etc)
- *Unsupervised* learning: only examples, no explicit feedback
 - goal to reveal structure in the observed data
- *Semi-supervised* learning: limited explicit feedback, mostly only examples
 - tries to improve predictions based on examples by making use of the additional “unlabeled” examples
- *Reinforcement* learning: delayed and partial feedback, no explicit guidance
 - goal to minimize the cost of a sequence of actions (policy)